Übung 13: Dynamische Webseiten mit JavaScript

Abgabedatum: 06 Feb. 2020, 12:00 Uhr

Aufgabe 1: CORS

3 Punkte

Normalerweise dürfen JavaScript Inhalte aus Sicherheitsgründen nur auf Inhalte zugreifen, die von der selben Domain kommen. In unserem Fall ist dies problematisch, da unsere Webanwendung Daten von `morgal.informatik.uni-ulm.de:8000/line/` und von dem bei Ihnen auf einem anderen Port laufenden Javaserver besorgen soll. Der andere Port reicht schon aus, um JavaScript Zugriffe nicht direkt zu erlauben. Informieren Sie sich über CORS.[1] Der Server unter `morgal.informatik.uni-ulm.de:8000/line/` ist bereits passend konfiguriert, passen Sie nun Ihren Javaserver (bzw. nehmen Sie die Musterlösung für Blatt 6 aus dem Moodle) an. Beim Aufruf der `/remote` URL sollen CORS Header so gesetzt werden, dass Anfragen über JavaScript erfolgreich sind.

Aufgabe 2: Hilfsfunktionen

2+2+2+2=8 Punkte

Wir wollen zunächst ein paar Hilfsfunktionen in einem Skript `js/util.js` implementieren. Sie können statt mit Ihrer eigenen Lösung auch mit der Musterlösung für Blatt 12 beginnen. Legen Sie die entsprechende Skriptdatei an und binden Sie sie in die Webseiten `index.php` und `stops.php` ein. Ausgenommen Event Handler (und ein wenig Code für die Kartenanzeige) soll in den PHP Dateien kein JavaScript Code existieren.

a) Implementieren Sie eine Funktion `load(url, onComplete, onError)` welche mittels Fetch API die Resource unter `url` anfragt. Im Fehlerfall soll die übergebene Funktion `onError` ohne Parameter ausgeführt werden, im Erfolgsfall soll die übergebene Funktion `onComplete` mit dem JSON Objekt, welches zurückgeliefert wurde, als Parameter ausgeführt werden.

b) Implementieren Sie Funktionen `getFavorite` und `setFavorite`, welche über die Local Storage API[a] die ID einer Haltestelle als Favorit speichert. In Falle von `getFavorite` soll `0` zurückgegeben werden wenn noch kein entsprechender Wert gespeichert wurde.

[a] `https://developer.mozilla.org/de/docs/Web/API/Window/localStorage`

c) Implementieren Sie die Funktion `loadStops(callback)`, welche von `morgal.informatik.uni-ulm.de:8000/line/stop` die Namen der Stops sowie ihre ID lädt und in eine JS Variable speichert. Die Funktion `callback` soll ohne Parameter aufgerufen werden, wenn der Ladevorgang abgeschlossen ist.

d) Implementieren Sie eine Funktion `toId(string)`, welche den Namen einer Haltestelle in ihre ID umsetzt und diese zurückgibt. Sie dürfen annehmen, dass die IDs der Stops mit `0` beginnen und dann um jeweils `1` inkrementiert werden.

Aufgabe 3: Dynamische Inhalte für die Hauptansicht

2+2+2=6 Punkte

Nun soll die Hauptansicht mit Leben befüllt werden. Legen Sie ein Skript `js/index.js` an und binden Sie

[1] `https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS`

es in die Hauptseite ein. Erneut soll nach Möglichkeit nur im Skript JavaScript enthalten sein, ausgenommen Event Handler. Wir implementieren zunächst nur die Befüllung der Favoritenbox. Die ID der entsprechenden Haltestelle lässt sich über `getFavorite()` ermitteln, konkrete Inhalte sind unter `morgal.informatik.uni-ulm.de:8000/line/stop/$id/departure` zu finden.

a) Implementieren Sie eine Funktion `loadFavorite()`, welche die Überschrift der Favoritenbox setzt z.B. indem sie die Werte von `morgal.informatik.uni-ulm.de:8000/line/stop/$id` liest.

b) Implementieren Sie eine Funktion `updateDepartures()`, welche die aktuellen Abfahrten an der Favoritenhaltestelle vom Backend lädt und in die Favoritenbox einfügt. Benutzen Sie die Funktion `setTimeout()`, um diese Funktion periodisch aufzurufen.

c) Implementieren Sie eine Funktion `initData()`, welche an geeigneter Stelle aufgerufen wird und die Webseite initial befüllt, sowie die periodische Aktualisierung der Abfahrten in der Favoritenbox anstößt.